Deep Graph Mapper: Seeing Graphs through the Neural Lens

Cristian Bodnar*, Cătălina Cangea*, Pietro Liò

Mila Graph Reading Group - 27 February 2020

UNIVERSITY OF CAMBRIDGE

Talk outline

- Motivation and background
- Deep Graph Mapper (DGM)
- Mapper-based PageRank pooling (MPR)
- Results
- Future work

Data & model visualisations = vital part of the research process, with several existing techniques (e.g. t-SNE, UMAP).

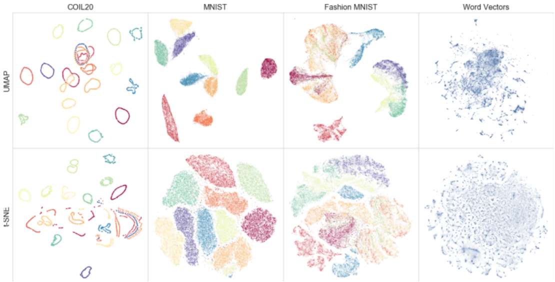


Image taken from McInnes et al. (2018).

These methods **assume stand-alone data points**, whereas graph-structured data contains relational information that would get thrown away!

This issue is addressed by graph neural networks, which are:

- **widely-used powerful mechanisms for learning abstract features** used in supervised/unsupervised settings and node/graph classification tasks...
- ...yet still **not useful when it comes to visualisation** purposes!

Motivation

Graph visualisations

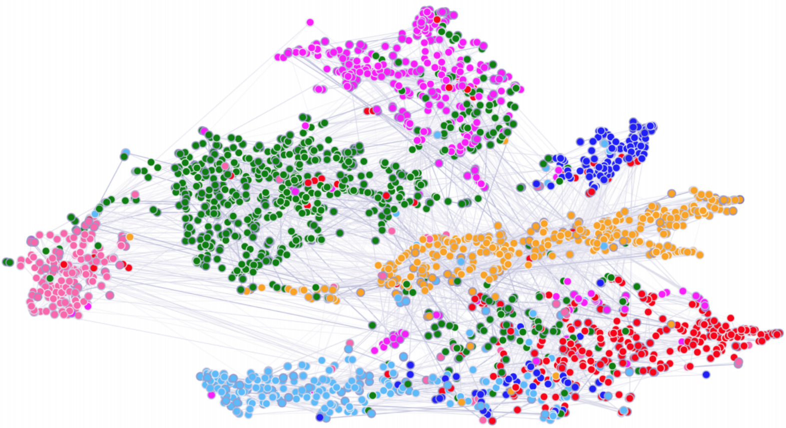


Image taken from Veličković et al. (2017).

Definition

An **open cover** of a topological space X is a collection of open sets $(U_i)_{i \in I}$, for some indexing set I, whose union includes X.

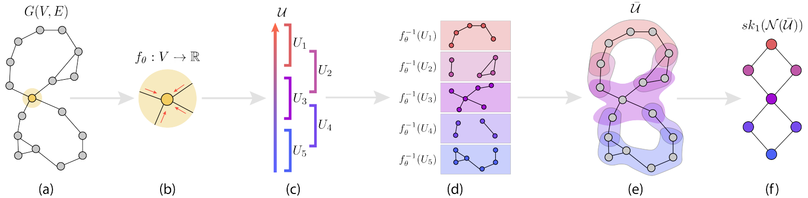
For example, an open cover for the real numbers could be $\{(-\infty, 0), (-2, 3), (1, \infty)\}$. Similarly, $\{\{v_1, v_2, v_3\}, \{v_4\}\}$ is an open cover for a set of vertices $\{v_1, v_2, v_3, v_4\}$.

Definition

Let X be a topological space, $f : X \to \mathbb{R}^d, d \geq 1$ a continuous function, and $\mathcal{U} = (U_i)_{i \in I}$ a cover of $\mathbb{R}^d$. Then, the **pull back cover** $f^*(\mathcal{U})$ of X induced by $(f, \mathcal{U})$ is the collection of open sets $f^{-1}(U_i), i \in I$, for some indexing set I, where by $f^{-1}(U_i)$ we denote the preimage of the set U_i.

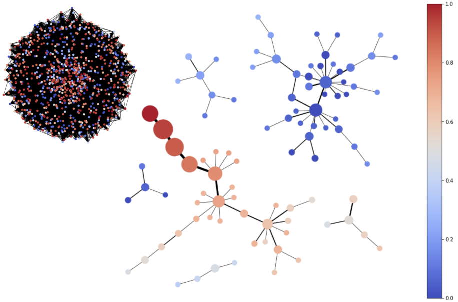
An example pull back cover is given in figures (d-e) on the next slide.

Deep Graph Mapper (DGM) merges **Mapper**, an algorithm from the field of **Topological Data Analysis** (TDA), with **Graph Neural Networks** (GNNs).

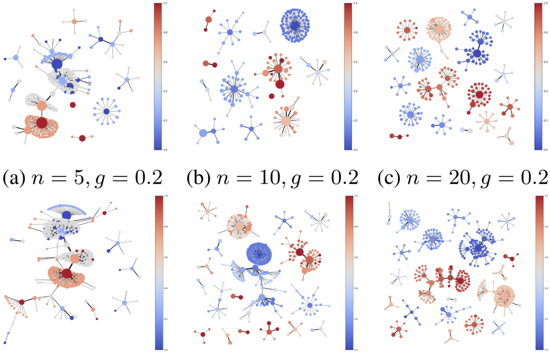


The lens function f_θ (b), represented here by a GNN, acts similarly to an optical lens. Its properties determine which features of the graph (a) should be highlighted in the visualisation (f). In this diagram, we assume f_θ outputs the height of each node in the plane of the diagram.

The graph contains spammers and non-spammers. Spammers are likely to be connected to many nodes in the graph, while non-spammers are connected to fewer nodes.



DGM can produce multi-resolution visualisations of the semantic relationships in the graph by adjusting the cover $\mathcal{U}$ defined by n (intervals) and g (overlap). They can be used to identify the persistent features of the graph.

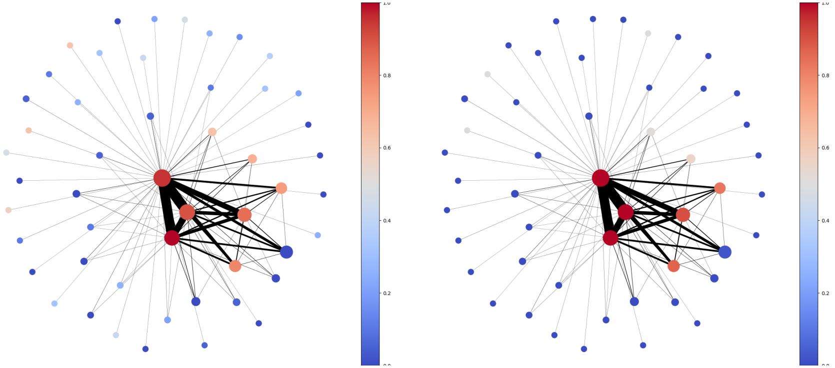


(a) $n = 5, g = 0.2$ (b) $n = 10, g = 0.2$ (c) $n = 20, g = 0.2$

(d) $n = 5, g = 0.4$ (e) $n = 10, g = 0.4$ (f) $n = 20, g = 0.4$

Structural Deep Graph Mapper

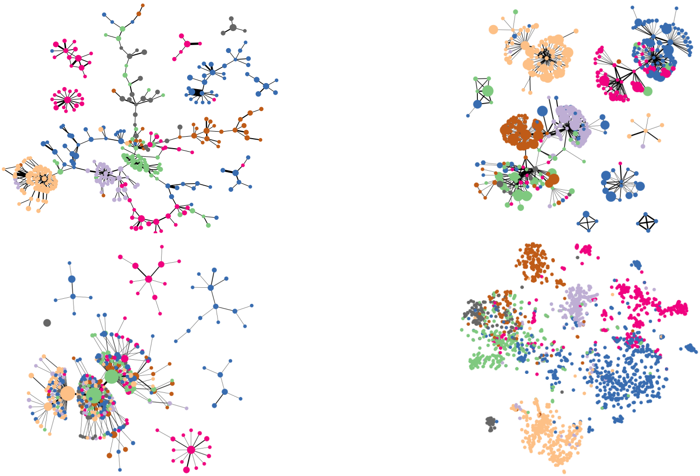
For most purposes, visualising structural relationships between clusters is more important than visualising semantic relationships. This is the purpose of SDGM which applies Mapper on the auxiliary graph (c) from the previous slide. Therefore, the thickness of the edges is proportional to the number of edges between clusters.



Qualitative Comparison

Cora

SDGM, DGM, Mapper with a graph density function and t-SNE.



Mapper for pooling

Mapper was primarily intended for visualisations, but we show that it is also suitable as a pooling mechanism. Theoretical results already support this (Hajij et al., 2018):

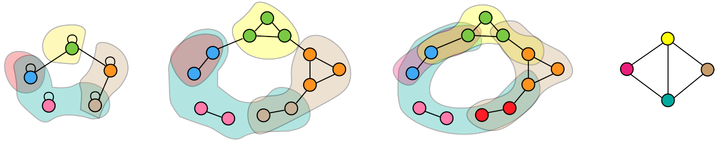
Proposition

Mapper is a generalisation of binary spectral clustering.

We prove an even stronger connection with pooling methods:

Proposition

Mapper is a generalisation of soft-cluster assignment pooling methods.



(a) (b) (c) (d)

Embedding network (L_E GCN layers before each pooling step):

$$\mathbf{X}_{l+1} = \sigma(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}_l\mathbf{W}_l) \tag{1}$$

Classification network output (after L_C GCN layers):

$$y = \mathrm{softmax}\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{X}_{L_C}\mathbf{W}_\mathrm{f} + \mathbf{b}_\mathrm{f}\right) \tag{2}$$

Mapper-based PageRank pooling (MPR)

Pooling layer

Use the lens $f : V \to \mathbb{R}$ (normalised PageRank (PR) of the nodes) to assign each node a real number in $[0, 1]$:

$$f(\mathbf{X}_{L_E})_i \triangleq \mathbf{PR}_i = \sum_{j \in N(i)} \frac{\mathbf{PR}_j}{|N(i)|} \tag{3}$$

Then, use the overlapping intervals cover $\mathcal{U}$ to build the soft cluster assignment matrix $\mathbf{S} \in \mathbb{R}^{|\mathrm{G}| \times |\mathrm{MG}|}$:

$$S_{ij} = \frac{\mathbb{I}_{i \in f^{-1}(U_j)}}{|\{U_k | i \in f^{-1}(U_k)\}|} \tag{4}$$

Mapper-based PageRank pooling (MPR)

Pooling layer

Finally, determine the pooled graph:

$$\begin{aligned} \mathbf{X}_{\mathrm{MG}} &= \mathbf{X}_{L_E}\mathbf{S}, \\ \mathbf{A}_{\mathrm{MG}} &= \mathbf{S}^T\mathbf{A}\mathbf{S}. \end{aligned} \tag{5}$$

MPR intuition: merge the (usually few) highly connected nodes in the graph and cluster the (typically many) dangling nodes with PageRank scores closer to zero.

Results

	MPR	Top-k	minCUT	DiffPool
D&D	$\mathbf{78.2 \pm 3.4}$	75.1 ± 2.2	77.6 ± 3.1	77.9 ± 2.4
Proteins	$\mathbf{75.2 \pm 2.2}$	74.8 ± 3.04	73.5 ± 2.9	74.2 ± 0.3
Collab	$\mathbf{81.5 \pm 1.0}$	75.0 ± 1.1	79.9 ± 0.8	81.3 ± 0.1
Reddit-B	86.3 ± 4.8	74.9 ± 7.4	$\mathbf{87.2 \pm 5.0}$	79.0 ± 1.1

Future work

- Probabilistic assignments based on location within intervals.
- Learnable lens and cover.
- Interactive visualisations.
- Alternative ways to visualise labels.
- Spatio-temporally evolving graphs.

Thank you!

Paper: arxiv.org/abs/2002.03864
Code: github.com/crisbodnar/dgm

@CristianBodnar
@catalinacangea